In the Claims:

Claim 1 (Currently amended): A computer program product embodied on computer readable media readable by a computing system in a computing environment, for enforcing security policy using style sheet processing, comprising:

computer-readable program code ~~means for obtaining~~ that is configured to obtain an input document;

computer-readable program code ~~means for obtaining~~ that is configured to obtain a Document Type Definition (DTD) that defines elements of said input document, wherein: (1) an attribute of at least one element defined in said DTD references one of a plurality of stored policy enforcement objects; (2) more than one of said references may reference a single stored policy enforcement object; and (3) each of said stored policy enforcement objects specifies a visibility policy for said referencing element or elements, said visibility policy identifying an encryption requirement for all elements having that visibility policy and a community whose members are authorized to view those elements;

computer-readable program code ~~means for applying~~ that is configured to apply one or more style sheets to said input document, thereby adding markup notation to each element of said input document for which said element definition in said DTD references one of said stored policy enforcement objects specifying a visibility policy with a non-null encryption requirement, resulting in creation of an interim transient document that indicates elements of said input document which are to be encrypted; ~~and~~

computer-readable program code ~~means for creating~~ that is configured to create an output document in which each element of said interim transient document for which markup

notation has been added is encrypted in a manner that enables a clerk process associated with a group that is a community member authorized to view that element to use key distribution material associated with the output document when decrypting the encrypted element.

Claim 2 (Currently amended): The computer program product according to Claim 1, further comprising computer-readable program code ~~means for rendering~~ that is configured to render said output document on a client device.

Claim 3 (Previously presented): The computer program product according to Claim 1, wherein said markup notation in said interim transient document comprises tags of a markup language.

Claim 4 (original): The computer program product according to Claim 1, wherein said input document is specified in an Extensible Markup Language (XML) notation.

Claim 5 (Previously presented): The computer program product according to Claim 4, wherein said output document is specified in said XML notation.

Claim 6 (Currently amended): The computer program product according to Claim 1, wherein said stored policy enforcement objects further comprise computer-readable program code ~~means for overriding~~ that is configured to override a method for evaluating said elements of said input document, and wherein said computer-readable program code ~~means~~

~~for applying~~ that is configured to apply said one or more style sheets further comprises computer-readable program code ~~means for invoking~~ that is configured to invoke said computer-readable program code ~~means for overriding~~ that is configured to override, thereby causing said markup notation to be added.

Claim 7 (original): The computer program product according to Claim 6, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

Claim 8 (Currently amended): The computer program product according to Claim 7, wherein said method is a value-of method of said XSL notation, and wherein said computer-readable program code ~~means for overriding~~ that is configured to override said value-of method is by subclassing said value-of method.

Claim 9 (Currently amended): The computer program product according to Claim 6, wherein:

said overriding method comprises:

computer-readable program code ~~means for generating~~ that is configured to generate said markup notation as encryption tags; and

computer-readable program code ~~means for inserting~~ that is configured to insert said generated encryption tags into said interim transient document to surround elements of said interim transient document for which said visibility policy of said elements in said input document have said non-null encryption requirement; and

said computer-readable program code ~~means for creating~~ that is configured to create said output document further comprises computer-readable program code ~~means for encrypting~~ that is configured to encrypt those elements surrounded by said inserted encryption tags.

Claim 10 (canceled).

Claim 11 (Previously presented): The computer program product according to Claim 1, wherein said encryption requirement further comprises specification of an encryption algorithm to be used when encrypting elements having that visibility policy.

Claim 12 (Previously presented): The computer program product according to Claim 1, wherein said encryption requirement further comprises specification of an encryption algorithm strength value to be used when encrypting elements having that visibility policy.

Claim 13 (Currently amended): The computer program product according to Claim 1, wherein said computer-readable program code ~~means for creating~~ that is configured to create said output document further comprises:

computer-readable program code ~~means for generating~~ that is configured to generate a distinct symmetric key for each unique one of said communities identified by said visibility policy in said stored policy objects for each of said elements of said input document; and

computer-readable program code ~~means for encrypting~~ that is configured to encrypt each of said distinct symmetric keys to create member-specific versions thereof, further comprising:

computer-readable program code ~~means for determining~~ that is configured to determine whether each of said members of said community for which said distinct symmetric key was generated is an individual or a group; and

computer-readable program code ~~means for encrypting~~ that is configured to encrypt a separate version of said distinct symmetric key for each determined individual and for a clerk process associated with each determined group.

Claim 14 (Currently amended): The computer program product according to Claim 13, wherein said computer-readable program code ~~means for encrypting~~ that is configured to encrypt a separate version of said distinct symmetric key creates one of said member-specific versions using, as input, a public key of one of said determined individuals or a public key of said clerk process.

Claim 15 (Previously presented): The computer program product according to Claim 1, wherein said encrypted elements in said created output document are encrypted using a cipher block chaining mode encryption process.

Claim 16 (Currently amended): The computer program product according to Claim 13, further comprising:

computer-readable program code ~~means for creating~~ that is configured to create a key class for each of said unique communities, wherein said key class is associated with each of said encrypted elements of said output document for which members of this unique community are authorized viewers, and wherein said key class comprises: (1) an encryption algorithm identifier and key length used when encrypting said associated encrypted elements; (2) an identifier of each of said members of said unique community; and (3) one of said member-specific versions of said encrypted symmetric key for each of said identified community members.

Claim 17 (Currently amended): The computer program product according to Claim 13, further comprising:

computer-readable program code ~~means for decrypting~~ that is configured to decrypt, for an individual user or process that is a member of one or more of said determined groups, only those encrypted elements in said output document for which any of said one or more of said determined groups is one of said authorized community members, further comprising:

computer-readable program code ~~means for expanding~~ that is configured to expand said determined groups to determine said individual users or processes that are group members in each of said expanded groups;

computer-readable program code ~~means for identifying~~ that is configured to identify one or more of said expanded groups of which said individual user or process is one of said group members;

computer-readable program code ~~means for decrypting~~ that is configured to decrypt, by said clerk process for each of said identified groups, said member-specific version of said symmetric key, thereby creating a decrypted key for each of said identified groups; and

computer-readable program code ~~means for decrypting~~ that is configured to decrypt selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said identified groups.

Claim 18 (Currently amended): The computer program product according to Claim 17, wherein:

said computer-readable program code ~~means for encrypting~~ that is configured to encrypt a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process;

said computer-readable program code ~~means for decrypting~~ that is configured to decrypt said member-specific version of said symmetric key further comprises:

computer-readable program code ~~means for contacting~~ that is configured to contact said clerk process, further comprising:

computer-readable program code ~~means for programmatically locating~~ that is configured to locate said clerk process; and

computer-readable program code ~~means for establishing~~ that is configured to establish a session between a client device used by said individual user or process and said clerk process;

computer-readable program code ~~means for digitally signing~~ that is configured to digitally sign said member-specific version by said individual user or process, thereby creating a first digital signature;

computer-readable program code ~~means for sending~~ that is configured to send said first digital signature and said member-specific version to said clerk process on said session;

computer-readable program code ~~means for receiving~~ that is configured to receive said sent first digital signature and said member-specific version by said clerk process;

computer-readable program code ~~means for verifying~~ that is configured to verify said first digital signature by said clerk process;

computer-readable program code ~~means for verifying~~ that is configured to verify, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

computer-readable program code ~~means for decrypting~~ that is configured to decrypt said member-specific version using a private key of said clerk process, wherein said private key is associated with said public key of said clerk process;

computer-readable program code ~~means for re-encrypting~~ that is configured to re-encrypt said decrypted member-specific version using a public key of said individual user or process, thereby creating a re-encrypted key;

computer-readable program code ~~means for digitally signing~~ that is configured to digitally sign said re-encrypted key by said clerk process, thereby creating a second digital signature;

computer-readable program code ~~means for returning~~ that is configured to return said second digital signature and said re-encrypted key from said clerk process to said client device on said session;

computer-readable program code ~~means for receiving~~ that is configured to receive said second digital signature and said re-encrypted key at said client device;

computer-readable program code ~~means for verifying~~ that is configured to verify said second digital signature at said client device; and

computer-readable program code ~~means, operable on said client device, for decrypting~~ that is configured to decrypt, on said client device, said received re-encrypted key using a private key of said individual user or process, creating said decrypted key; and

said computer-readable program code ~~means for decrypting~~ that is configured to decrypt selected ones of said encrypted elements in said output document is executed at said client device using said decrypted key.

Claim 19 (Currently amended): The computer program product according to Claim 13, further comprising:

computer-readable program code ~~means for decrypting~~ that is configured to decrypt, for an individual user or process that is a member of one of said determined groups, only those encrypted elements in said output document for which any of said one or more of said determined groups is one of said authorized community members, further comprising:

computer-readable program code ~~means for expanding~~ that is configured to expand said determined groups to determine said individual users or processes that are group members in each of said expanded groups;

computer-readable program code ~~means for identifying~~ that is configured to identify one or more of said expanded groups of which said individual user or process is one of said group members; and

computer-readable program code ~~means for decrypting~~ that is configured to decrypt selected ones of said encrypted elements in said output document, wherein said selected ones of said encrypted elements are those which were encrypted for one of said identified groups .

Claim 20 (Currently amended): The computer program product according to Claim 19, further comprising:

computer-readable program code ~~means for contacting~~ that is configured to contact said clerk process, further comprising:

computer-readable program code ~~means for programmatically locating~~ that is configured to locate said clerk process; and

computer-readable program code ~~means for establishing~~ that is configured to establish a mutually-authenticated secure session between a client device used by said individual user or process and said clerk process; and wherein:

said computer-readable program code ~~means for encrypting~~ that is configured to encrypt a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process; and

said computer-readable program code ~~means for decrypting~~ that is configured to decrypt selected ones of said encrypted elements in said output document further comprises:

computer-readable program code ~~means for locating~~ that is configured to locate said member-specific version of said symmetric key which was encrypted using said public key of said clerk process, wherein said clerk process is associated with a group of which said individual user or process is a group member;

computer-readable program code ~~means for sending~~ that is configured to send said located member-specific version to said clerk process, along with an element encrypted with said member-specific version, on said secure session;

computer-readable program code ~~means for receiving~~ that is configured to receive said sent member-specific version and said element by said clerk process;

computer-readable program code ~~means for verifying~~ that is configured to verify, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

computer-readable program code ~~means for decrypting~~ that is configured to decrypt said member-specific version using a private key of said clerk process;

computer-readable program code ~~means for decrypting~~ that is configured to decrypt said element using said decrypted member-specific version; and

computer-readable program code ~~means for returning~~ that is configured to return said decrypted element from said clerk process to said client device on said secure session.

Claim 21 (Currently amended): The computer program product according to Claim 16, wherein:

said computer-readable program code ~~means for encrypting~~ that is configured to encrypt a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process; and further comprising:

computer-readable program code ~~means for contacting~~ that is configured to contact said clerk process, further comprising:

computer-readable program code ~~means for programmatically locating~~ that is configured to locate said clerk process; and

computer-readable program code ~~means for establishing~~ that is configured to establish a mutually-authenticated secure session between a client device used by said individual user or process and said clerk process;

computer-readable program code ~~means for decrypting~~ that is configured to decrypt, for an individual user or process that is a member of one of said determined groups,

only those encrypted elements in said output document for which any of said one or more of said determined groups is one of said authorized community members, further comprising:

computer-readable program code ~~means for expanding~~ that is configured to expand said determined groups to determine said individual users or processes that are group members in each of said expanded groups;

computer-readable program code ~~means for identifying~~ that is configured to identify one or more of said key classes which identify said individual user or process as one of said group members;

computer-readable program code ~~means for decrypting~~ that is configured to decrypt, for each of said determined key classes, said member-specific version of said symmetric key in said key class which was encrypted using said public key of said clerk process , wherein said computer-readable program code ~~means for decrypting~~ that is configured to decrypt uses a private key of said clerk process , thereby creating a decrypted key; and

computer-readable program code ~~means for decrypting~~ that is configured to decrypt selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for said key class.

Claim 22 (Currently amended): The computer program product according to Claim 17, wherein:

said computer-readable program code ~~means for decrypting~~ that is configured to decrypt said member-specific version further comprises:

computer-readable program code ~~means for locating~~ that is configured to locate said clerk process; and

computer-readable program code ~~means for establishing~~ that is configured to establish a mutually-authenticated secure session between said client device and said clerk process;

computer-readable program code ~~means for sending~~ that is configured to send said member-specific version to said clerk process on said secure session;

computer-readable program code ~~means for receiving~~ that is configured to receive said sent member-specific version by said clerk process;

computer-readable program code ~~means for verifying~~ that is configured to verify, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

computer-readable program code ~~means for decrypting~~ that is configured to decrypt said member-specific version using a private key of said clerk process ;

computer-readable program code ~~means for returning~~ that is configured to return said decrypted member-specific version from said clerk process to said client device on said secure session; and

computer-readable program code ~~means for receiving~~ that is configured to receive said decrypted member-specific version at said client device; and

said computer-readable program code ~~means for decrypting~~ that is configured to decrypt selected ones of said encrypted elements in said output document is executed at said client device using said received decrypted member-specific version.

Claim 23 (Currently amended): The computer program product according to Claim 17, Claim 21, or Claim 22, further comprising computer-readable program code ~~means for substituting~~ that is configured to substitute a predetermined text message for any encrypted elements in said output document which cannot be decrypted for said individual user or process.

Claim 24 (Currently amended): The computer program product according to Claim 19, further comprising:

computer-readable program code ~~means for contacting~~ that is configured to contact said clerk process, further comprising:

computer-readable program code ~~means for programmatically locating~~ that is configured to locate said clerk process; and

computer-readable program code ~~means for establishing~~ that is configured to establish a session between a client device used by said individual user or process and said clerk process; and wherein:

said computer-readable program code ~~means for encrypting~~ that is configured to encrypt a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process; and

said computer-readable program code ~~means for decrypting~~ that is configured to decrypt selected ones of said encrypted elements in said output document further comprises:

computer-readable program code ~~means for locating~~ that is configured to locate said member-specific version of said r symmetric key which was encrypted using said public key of said clerk process, wherein said clerk process is associated with a group of which said individual user or process is a group member;

computer-readable program code ~~means for digitally signing~~ that is configured to digitally sign, by said individual user or process, said located version and an element encrypted with said member-specific version, thereby creating a first digital signature;

computer-readable program code ~~means for sending~~ that is configured to send said first digital signature, said located member-specific version, and said element to said clerk process on said session;

computer-readable program code ~~means for receiving~~ that is configured to receive said sent first digital signature, said member-specific version, and said element by said clerk process;

computer-readable program code ~~means for verifying~~ that is configured to verify said first digital signature by said clerk process;

computer-readable program code ~~means for verifying~~ that is configured to verify, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

computer-readable program code ~~means for decrypting~~ that is configured to decrypt said member-specific version using a private key of said clerk process;

computer-readable program code ~~means for decrypting~~ that is configured to decrypt said element using said decrypted member-specific version;

computer-readable program code ~~means for re-encrypting~~ that is configured to re-encrypt said decrypted element using a public key of said individual user or process, thereby creating a re-encrypted element;

computer-readable program code ~~means for digitally signing~~ that is configured to digitally sign said re-encrypted element by said clerk process, thereby creating a second digital signature;

computer-readable program code ~~means for returning~~ that is configured to return said second digital signature and said re-encrypted element from said clerk process to said client device on said session;

computer-readable program code ~~means for receiving~~ that is configured to receive said second digital signature and said re-encrypted element at said client device; and

computer-readable program code ~~means for verifying~~ that is configured to verify said second digital signature by said individual user or process.

Claim 25 (original): The computer program product according to Claim 1, wherein said DTD is replaced by a schema.

Claim 26 (Previously presented): The computer program product according to Claim 1, wherein said encryption requirement further comprises specification of an encryption key length.

Claim 27 (original): The computer program product according to Claim 9, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.

Claim 28 (Previously presented): A system for enforcing security policy using style sheet processing in a computing environment, comprising:

means for obtaining an input document;

means for obtaining a Document Type Definition (DTD) that defines elements of said input document, wherein: (1) an attribute of at least one element defined in said DTD references one of a plurality of stored policy enforcement objects; (2) more than one of said references may reference a single stored policy enforcement object; and (3) each of said stored policy enforcement objects specifies a visibility policy for said referencing element or elements, said visibility policy identifying an encryption requirement for all elements having that visibility policy and a community whose members are authorized to view those elements;

means for applying one or more style sheets to said input document, thereby adding markup notation to each element of said input document for which said element definition in said DTD references one of said stored policy enforcement objects specifying a visibility policy with a non-null encryption requirement, resulting in creation of an interim transient document that indicates elements of said input document which are to be encrypted; and

means for creating an output document in which each element of said interim transient document for which markup notation has been added is encrypted in a manner that

enables a clerk process associated with a group that is a community member authorized to view that element to use key distribution material associated with the output document when decrypting the encrypted element.

Claim 29 (Previously presented): The system according to Claim 28, further comprising means for rendering said output document on a client device.

Claim 30 (Previously presented): The system according to Claim 28, wherein said markup notation in said interim transient document comprises tags of a markup language.

Claim 31 (original): The system according to Claim 28, wherein said input document is specified in an Extensible Markup Language (XML) notation.

Claim 32 (Previously presented): The system according to Claim 31, wherein said output document is specified in said XML notation.

Claim 33 (Previously presented): The system according to Claim 28, wherein said stored policy enforcement objects further comprise means for overriding a method for evaluating said elements of said input document, and wherein said means for applying said one or more style sheets further comprises means for invoking said means for overriding, thereby causing said markup notation to be added.

Claim 34 (original): The system according to Claim 33, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

Claim 35 (original): The system according to Claim 34, wherein said method is a value-of method of said XSL notation, and wherein said means for overriding said value-of method is by subclassing said value-of method.

Claim 36 (Previously presented): The system according to Claim 33, wherein:

said overriding method comprises:

means for generating said markup notation as encryption tags; and

means for inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document for which said visibility policy of said elements in said input document have said non-null encryption requirement; and

said means for creating said output document further comprises means for encrypting those elements surrounded by said inserted encryption tags.

Claim 37 (canceled).

Claim 38 (Previously presented): The system according to Claim 28, wherein said encryption requirement further comprises specification of an encryption algorithm to be used when encrypting elements having that visibility policy.

Claim 39 (Previously presented): The system according to Claim 28, wherein said encryption requirement further comprises specification of an encryption algorithm strength value to be used when encrypting elements having that visibility policy.

Claim 40 (Previously presented): The system according to Claim 28, wherein said means for creating said output document further comprises:

means for generating a distinct symmetric key for each unique one of said communities identified by said visibility policy in said stored policy objects for each of said elements of said input document; and

means for encrypting each of said distinct symmetric keys to create member-specific versions thereof, further comprising:

means for determining whether each of said members of said community for which said distinct symmetric key was generated is an individual or a group; and

means for encrypting a separate version of said distinct symmetric key for each determined individual and for a clerk process associated with each determined group.

Claim 41 (Previously presented): The system according to Claim 40, wherein said means for encrypting a separate version of said distinct symmetric key creates one of said member-specific versions using, as input, a public key of one of said determined individuals or a public key of said clerk process.

Claim 42 (Previously presented): The system according to Claim 28, wherein said encrypted elements in said created output document are encrypted using a cipher block chaining mode encryption process.

Claim 43 (Previously presented): The system according to Claim 40, further comprising:

means for creating a key class for each of said unique communities, wherein said key class is associated with each of said encrypted elements of said output document for which members of this unique community are authorized viewers, and wherein said key class comprises: (1) an encryption algorithm identifier and key length used when encrypting said associated encrypted elements; (2) an identifier of each of said members of said unique community; and (3) one of said member-specific versions of said encrypted symmetric key for each of said identified community members.

Claim 44 (Previously presented): The system according to Claim 40, further comprising:

means for decrypting, for an individual user or process that is a member of one or more of said determined groups, only those encrypted elements in said output document for which any of said one or more of said determined groups is one of said authorized community members, further comprising:

means for expanding said determined groups to determine said individual users or processes that are group members in each of said expanded groups;

means for identifying one or more of said expanded groups of which said individual user or process is one of said group members;

means for decrypting, by said clerk process for each of said identified groups, said member-specific version of said symmetric key, thereby creating a decrypted key for each of said identified groups; and

means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said identified groups.

Claim 45 (Previously presented): The system according to Claim 44, wherein:

said means for encrypting a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process;

said means for decrypting said member-specific version of said symmetric key further comprises:

means for contacting said clerk process, further comprising:

means for programmatically locating said clerk process; and

means for establishing a session between a client device used by said individual user or process and said clerk process;

means for digitally signing said member-specific version by said individual user or process, thereby creating a first digital signature;

means for sending said first digital signature and said member-specific version to said clerk process on said session;

means for receiving said sent first digital signature and said member-specific version by said clerk process;

means for verifying said first digital signature by said clerk process;

means for verifying, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

means for decrypting said member-specific version using a private key of said clerk process, wherein said private key is associated with said public key of said clerk process;

means for re-encrypting said decrypted member-specific version using a public key of said individual user or process, thereby creating a re-encrypted key;

means for digitally signing said re-encrypted key by said clerk process, thereby creating a second digital signature;

means for returning said second digital signature and said re-encrypted key from said clerk process to said client device on said session;

means for receiving said second digital signature and said re-encrypted key at said client device;

means for verifying said second digital signature at said client device; and

means, operable on said client device, for decrypting said received re-encrypted key using a private key of said individual user or process, creating said decrypted key; and

said means for decrypting selected ones of said encrypted elements in said output document is executed at said client device using said decrypted key.

Claim 46 (Previously presented): The system according to Claim 40, further comprising:

means for decrypting, for an individual user or process that is a member of one of said determined groups, only those encrypted elements in said output document for which any of said one or more of said determined groups is one of said authorized community members, further comprising:

means for expanding said determined groups to determine said individual users or processes that are group members in each of said expanded groups;

means for identifying one or more of said expanded groups of which said individual user or process is one of said group members; and

means for decrypting selected ones of said encrypted elements in said output document, wherein said selected ones of said encrypted elements are those which were encrypted for one of said identified groups .

Claim 47 (Previously presented): The system according to Claim 46, further comprising:

means for contacting said clerk process, further comprising:

means for programmatically locating said clerk process; and

means for establishing a mutually-authenticated secure session between a client device used by said individual user or process and said clerk process; and wherein:

said means for encrypting a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process; and

said means for decrypting selected ones of said encrypted elements in said output document further comprises:

means for locating said member-specific version of said symmetric key which was encrypted using said public key of said clerk process, wherein said clerk process is associated with a group of which said individual user or process is a group member;

means for sending said located member-specific version to said clerk process, along with an element encrypted with said member-specific version, on said secure session;

means for receiving said sent member-specific version and said element by said clerk process;

means for verifying, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

means for decrypting said member-specific version using a private key of said clerk process; means for decrypting said element using said decrypted member-specific version; and

means for returning said decrypted element from said clerk process to said client device on said secure session.

Claim 48 (Previously presented): The system according to Claim 43, wherein: said means for encrypting a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process; and further comprising:

means for contacting said clerk process, further comprising:

means for programmatically locating said clerk process; and

means for establishing a mutually-authenticated secure session between a client device used by said individual user or process and said clerk process;

means for decrypting, for an individual user or process that is a member of one of said determined groups, only those encrypted elements in said output document for which any of said one or more of said determined groups is one of said authorized community members, further comprising: means for expanding said determined groups to determine said individual users or processes that are group members in each of said expanded groups;

means for identifying one or more of said key classes which identify said individual user or process as one of said group members;

means for decrypting, for each of said determined key classes, said member-specific version of said symmetric key in said key class which was encrypted using said public key of said clerk process , wherein said means for decrypting uses a private key of said clerk process , thereby creating a decrypted key; and

means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for said key class.

Claim 49 (Previously presented): The system according to Claim 44, wherein:

said means for decrypting said member-specific version further comprises:

means for locating said clerk process; and

means for establishing a mutually-authenticated secure session between said client device and said clerk process;

means for sending said member-specific version to said clerk process on said secure session;

means for receiving said sent member-specific version by said clerk process;

means for verifying, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

means for decrypting said member-specific version using a private key of said clerk process ;

means for returning said decrypted member-specific version from said clerk process to said client device on said secure session; and

means for receiving said decrypted member-specific version at said client device; and

said means for decrypting selected ones of said encrypted elements in said output document is executed at said client device using said received decrypted member-specific version.

Claim 50 (Previously presented): The system according to Claim 44, Claim 48, or Claim 49, further comprising means for substituting a predetermined text message for any encrypted elements in said output document which cannot be decrypted for said individual user or process.

Claim 51 (Previously presented): The system according to Claim 46, further comprising:

means for contacting said clerk process, further comprising:

means for programmatically locating said clerk process; and

means for establishing a session between a client device used by said individual user or process and said clerk process; and wherein:

said means for encrypting a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process; and

said means for decrypting selected ones of said encrypted elements in said output document further comprises:

means for locating said member-specific version of said r symmetric key which was encrypted using said public key of said clerk process, wherein said clerk process is associated with a group of which said individual user or process is a group member; means for digitally signing, by said individual user or process, said located version and an element encrypted with said member-specific version, thereby creating a first digital signature;

means for sending said first digital signature, said located member-specific version, and said element to said clerk process on said session;

means for receiving said sent first digital signature, said member-specific version, and said element by said clerk process;

means for verifying said first digital signature by said clerk process;

means for verifying, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

means for decrypting said member-specific version using a private key of said clerk process; means for decrypting said element using said decrypted member-specific version;

means for re-encrypting said decrypted element using a public key of said individual user or process, thereby creating a re-encrypted element;

means for digitally signing said re-encrypted element by said clerk process, thereby creating a second digital signature;

means for returning said second digital signature and said re-encrypted element from said clerk process to said client device on said session;

means for receiving said second digital signature and said re-encrypted element at said client device; and

means for verifying said second digital signature by said individual user or process.

Claim 52 (original): The system according to Claim 28, wherein said DTD is replaced by a schema.

Claim 53 (Previously presented): The system according to Claim 28, wherein said encryption requirement further comprises specification of an encryption key length.

Claim 54 (original): The system according to Claim 36, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.

Claim 55 (Currently amended): A method for enforcing security policy using style sheet processing, comprising ~~the steps of~~:

providing an input document;

providing a Document Type Definition (DTD) that defines elements of said input document, wherein: (1) an attribute of at least one element defined in said DTD references one of a plurality of stored policy enforcement objects; (2) more than one of said references may reference a single stored policy enforcement object; and (3) each of said stored policy enforcement objects specifies a visibility policy for said referencing element or elements, said visibility policy identifying an encryption requirement for all elements having that visibility policy and a community whose members are authorized to view those elements;

applying one or more style sheets to said input document, thereby adding markup notation to each element of said input document for which said element definition in said DTD references one of said stored policy enforcement objects specifying a visibility policy with a non-null encryption requirement, resulting in creation of an interim transient document that indicates elements of said input document which are to be encrypted; and

creating an output document in which each element of said interim transient document for which markup notation has been added is encrypted in a manner that enables a clerk process associated with a group that is a community member authorized to view that element

to use key distribution material associated with the output document when decrypting the encrypted element.

Claim 56 (Currently amended): The method according to Claim 55, further comprising ~~the step of~~ rendering said output document on a client device.

Claim 57 (Previously presented): The method according to Claim 55, wherein said markup notation in said interim transient document comprises tags of a markup language.

Claim 58 (original): The method according to Claim 55, wherein said input document is specified in an Extensible Markup Language (XML) notation.

Claim 59 (Previously presented): The method according to Claim 58, wherein said output document is specified in said XML notation.

Claim 60 (Currently amended): The method according to Claim 55, wherein said stored policy enforcement objects further comprise executable code for overriding a method for evaluating said elements of said input document, and wherein said applying said one or more style sheets ~~step~~ further comprises overriding said method for evaluating by invoking said executable code, thereby causing said markup notation to be added.

Claim 61 (original): The method according to Claim 60, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

Claim 62 (Currently amended): The method according to Claim 61, wherein said method is a value-of method of said XSL notation, and wherein said ~~step of~~ overriding said value-of method is by subclassing said value-of method.

Claim 63 (Currently amended): The method according to Claim 60, wherein:

said ~~step of~~ overriding further comprises ~~the steps of~~:

generating said markup notation as encryption tags; and

inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document for which said visibility policy of said elements in said input document have said non-null encryption requirement; and

said ~~step of~~ creating said output document further comprises ~~the step of~~ encrypting those elements surrounded by said inserted encryption tags.

Claim 64 (canceled).

Claim 65 (Previously presented): The method according to Claim 55, wherein said encryption requirement further comprises specification of an encryption algorithm to be used when encrypting elements having that visibility policy.

Claim 66 (Previously presented): The method according to Claim 2558, wherein said encryption requirement further comprises specification of an encryption algorithm strength value to be used when encrypting elements having that visibility policy.

Claim 67 (Currently amended): The method according to Claim 55, wherein said ~~step of~~ creating said output document further comprises ~~the steps of~~:

generating a distinct symmetric key for each unique one of said communities identified by said visibility policy in said stored policy objects for each of said elements of said input document; and

encrypting each of said distinct symmetric keys to create member-specific versions thereof, further comprising ~~the steps of~~:

determining whether each of said members of said community for which said distinct symmetric key was generated is an individual or a group; and

encrypting a separate version of said distinct symmetric key for each determined individual and for a clerk process associated with each determined group.

Claim 68 (Currently amended): The method according to Claim 67, wherein said ~~step of~~ encrypting a separate version of said distinct symmetric key creates one of said member-specific versions using, as input, a public key of one of said determined individuals or a public key of said clerk process.

Claim 69 (Previously presented): The method according to Claim 55, wherein said encrypted elements in said created output document are encrypted using a cipher block chaining mode encryption process.

Claim 70 (Currently amended): The method according to Claim 67, further comprising ~~the step of~~:

creating a key class for each of said unique communities, wherein said key class is associated with each of said encrypted elements of said output document for which members of this unique community are authorized viewers, and wherein said key class comprises: (1) an encryption algorithm identifier and key length used when encrypting said associated encrypted elements; (2) an identifier of each of said members of said unique community; and (3) one of said member-specific versions of said encrypted symmetric key for each of said identified community members.

Claim 71 (Currently amended): The method according to Claim 67, further comprising ~~the step of~~:

decrypting, for an individual user or process that is a member of one or more of said determined groups, only those encrypted elements in said output document for which any of said one or more of said determined groups is one of said authorized community members, further comprising ~~the steps of~~:

expanding said determined groups to determine said individual users or processes that are group members in each of said expanded groups;

identifying one or more of said expanded groups of which said individual user or process is one of said group members;

decrypting, by said clerk process for each of said identified groups, said member-specific version of said symmetric key, thereby creating a decrypted key for each of said identified groups; and

decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said identified groups.

Claim 72 (Currently amended): The method according to Claim 71, wherein:

said ~~step of~~ encrypting a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process;

said ~~step of~~ decrypting said member-specific version of said symmetric key further comprises ~~the steps of~~:

contacting said clerk process, further comprising ~~the steps of~~:

programmatically locating said clerk process; and

establishing a session between a client device used by said individual user or process and said clerk process;

digitally signing said member-specific version by said individual user or process, thereby creating a first digital signature;

sending said first digital signature and said member-specific version to said clerk process on said session;

receiving said sent first digital signature and said member-specific version by said clerk process;

verifying said first digital signature by said clerk process;

verifying, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

decrypting said member-specific version using a private key of said clerk process, wherein said private key is associated with said public key of said clerk process;

re-encrypting said decrypted member-specific version using a public key of said individual user or process, thereby creating a re-encrypted key;

digitally signing said re-encrypted key by said clerk process, thereby creating a second digital signature;

returning said second digital signature and said re-encrypted key from said clerk process to said client device on said session;

receiving said second digital signature and said re-encrypted key at said client device;

verifying said second digital signature at said client device; and

decrypting, at said client device, said received re-encrypted key using a private key of said individual user or process, creating said decrypted key; and

said ~~step of~~ decrypting selected ones of said encrypted elements in said output document is executed at said client device using said decrypted key.

Claim 73 (Currently amended): The method according to Claim 67, further comprising ~~the step of~~:

decrypting, for an individual user or process that is a member of one of said determined groups, only those encrypted elements in said output document for which any of said one or more of said determined groups is one of said authorized community members, further comprising ~~the steps of~~:

expanding said determined groups to determine said individual users or processes that are group members in each of said expanded groups;

identifying one or more of said expanded groups of which said individual user or process is one of said group members; and

decrypting selected ones of said encrypted elements in said output document, wherein said selected ones of said encrypted elements are those which were encrypted for one of said identified groups .

Claim 74 (Currently amended): The method according to Claim 73, further comprising ~~the step of~~ contacting said clerk process, further comprising ~~the steps of~~:

programmatically locating said clerk process; and

establishing a mutually-authenticated secure session between a client device used by said individual user or process and said clerk process; and wherein:

said ~~step of~~ encrypting a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process; and

said ~~step of~~ decrypting selected ones of said encrypted elements in said output document further comprises ~~the steps of~~:

locating said member-specific version of said symmetric key which was encrypted using said public key of said clerk process, wherein said clerk process is associated with a group of which said individual user or process is a group member;

sending said located member-specific version to said clerk process, along with an element encrypted with said member-specific version, on said secure session;

receiving said sent member-specific version and said element by said clerk process;

verifying, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

decrypting said member-specific version using a private key of said clerk process; decrypting said element using said decrypted member-specific version; and

returning said decrypted element from said clerk process to said client device on said secure session.

Claim 75 (Currently amended): The method according to Claim 70, wherein:

said ~~step of~~ encrypting a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process; and further comprising ~~the steps of~~:

contacting said clerk process, further comprising ~~the steps of~~:

programmatically locating said clerk process; and

establishing a mutually-authenticated secure session between a client device used by said individual user or process and said clerk process;

~~step of~~ decrypting, for an individual user or process that is a member of one of said determined groups, only those encrypted elements in said output document for which any of said one or more of said determined groups is one of said authorized community members, further comprising ~~the steps of~~:

expanding said determined groups to determine said individual users or processes that are group members in each of said expanded groups;

identifying one or more of said key classes which identify said individual user or process as one of said group members;

decrypting, for each of said determined key classes, said member-specific version of said symmetric key in said key class which was encrypted using said public key of said clerk process , wherein said ~~step of~~ decrypting uses a private key of said clerk process , thereby creating a decrypted key; and

decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for said key class.

Claim 76 (Currently amended): The method according to Claim 71, wherein:

said ~~step of~~ decrypting said member-specific version further comprises ~~the steps of~~:

locating said clerk process; and

establishing a mutually-authenticated secure session between said client device and said clerk process;

sending said member-specific version to said clerk process on said secure session;

receiving said sent member-specific version by said clerk process;

verifying, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

decrypting said member-specific version using a private key of said clerk process;

returning said decrypted member-specific version from said clerk process to said client device on said secure session; and

receiving said decrypted member-specific version at said client device; and

said ~~step of~~ decrypting selected ones of said encrypted elements in said output document is executed at said client device using said received decrypted member-specific version.

Claim 77 (Currently amended): The method according to Claim 71, Claim 75, or Claim 76, further comprising ~~the step of~~ substituting a predetermined text message for any encrypted elements in said output document which cannot be decrypted for said individual user or process.

Claim 78 (Currently amended): The method according to Claim 73, further comprising ~~the steps of~~:

contacting said clerk process, further comprising ~~the steps of~~:

programmatically locating said clerk process; and

establishing a session between a client device used by said individual user or process and said clerk process; and wherein:

said ~~step of~~ encrypting a separate version uses a public key of said clerk process as input when creating said member-specific version for said clerk process; and

said ~~step of~~ decrypting selected ones of said encrypted elements in said output document further comprises ~~the steps of~~:

locating said member-specific version of said r symmetric key which was encrypted using said public key of said clerk process, wherein said clerk process is associated with a group of which said individual user or process is a group member; digitally signing, by said individual user or process, said located version and an element encrypted with said member-specific version, thereby creating a first digital signature;

sending said first digital signature, said located member-specific version, and said element to said clerk process on said session;

receiving said sent first digital signature, said member-specific version, and said element by said clerk process;

verifying said first digital signature by said clerk process;

verifying, by said clerk process, that said individual user or process is one of said members of said identified group associated with said member-specific version;

decrypting said member-specific version using a private key of said clerk process; decrypting said element using said decrypted member-specific version;

re-encrypting said decrypted element using a public key of said individual user or process, thereby creating a re-encrypted element;

digitally signing said re-encrypted element by said clerk process, thereby creating a second digital signature;

returning said second digital signature and said re-encrypted element from said clerk process to said client device on said session;

receiving said second digital signature and said re-encrypted element at said client device; and

verifying said second digital signature by said individual user or process.

Claim 79 (original): The method according to Claim 55, wherein said DTD is replaced by a schema.

Claim 80 (Previously presented): The method according to Claim 55, wherein said encryption requirement further comprises specification of an encryption key length.

Claim 81 (original): The method according to Claim 63, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.